

IN THE CLAIMS

Please amend the claims as follows:

1.-7. (Canceled)

8. (Currently Amended) A computer-implemented method to manage interactions between applications and a data store and to process on a computer to perform the method, comprising:

receiving a query for a data store and an identifier for an application, wherein the application when executed seeks to process results returned from and produced by executing the query and seeks to update the data store with application data, wherein the application data is produced in response to the application processing the results of the query;

concurrently executing multiple instances of the ~~[[an]]~~ application associated with the identifier on multiple processing nodes within a network to achieve parallel processing for the multiple instances of the application;

concurrently processing the query to acquire the results on behalf of the multiple instances of the application and producing the results that are then streamed to ~~housed in~~ a plurality of application queues residing on a plurality of the processing nodes as the results are acquired, each application queue having different portions of the results; and

concurrently providing the results to each of the instances of the application from the application queues so that the instances can produce the application data from the results that are streamed to load queues for a single ~~[[and]]~~ update to the data store with all the application data, which is to be subsequently accessed from the data store, and wherein the update to the data store is done after each instance of the applications finishes its processing and has streamed its application data to the load queues.

9. (Previously Presented) The method of claim 8 further comprising:

concurrently housing the application data in one or more load queues residing on one or more of the processing nodes; and

concurrently populating one or more tables residing on the processing nodes with the application data from the one or more load queues.

10. (Original) The method of claim 9 further comprising merging the one or more tables into the data store.

11. (Original) The method of claim 8 wherein the currently initiating further includes determining a total number of the applications to initiate based on configuration data.

12. (Original) The method of claim 11 wherein the currently initiating further includes determining which of a number of the applications that are to be initiated on which of a number of the processing nodes based on the configuration data.

13. (Previously Presented) The method of claim 8 further comprising concurrently synchronizing the application queues and the load queues on the multiple processing nodes when at least some of the processing nodes lack one of the application queues or one of one or more load queues.

14. (Original) The solution template system of claim 13 wherein the concurrently synchronizing further includes establishing socket based communications between the multiple processing nodes with a Transmission Control Protocol/Internet Protocol (TCP/IP).

15. (Currently Amended) A data store application management system implemented in a processing and memory device, comprising:

application queues for servicing applications with results of a query to a data store, wherein the application queues reside within memory of the device and receive the results via streaming when the query executes, and wherein each application queue includes a different portion of the results;

load queues for housing application data produced by the applications that process the results from the application queues, wherein the load queues received the application data via

streaming from the applications and the applications are multiple instances of one another, and wherein the load queues reside within memory of the device and receive the application data when the applications execute in parallel with one another as ~~application~~ instances of one another on multiple node devices of a network; and

a merge utility for merging the application data into a data store table, wherein the merge utility executes on the device and merges the load queues into the data store table to update the data store with the application data that is to be subsequently accessed from the data store, and wherein the update to the data store with the application data occurs once and after all the applications have finished processing and streamed their ~~produced~~ the application data to the load queues.

16. (Previously Presented) The system of claim 15 further comprising a configuring utility for determining a total number of the applications.

17. (Previously Presented) The system of claim 15, wherein the configuring utility initiates a number of the applications, the application queues, and the load queues on separate processing nodes.

18. (Previously Presented) The system of claim 15, wherein each of the applications concurrently processes the results and produces different portions of the application data.

19. (Previously Presented) The system of claim 18, wherein each of the application queues and each of the load queues concurrently update while the applications process.

20. (Currently Amended) A data store implemented in a processing and memory device and accessible over a network, comprising:

temporary tables that temporarily house application data produced from concurrently processing multiple instances of an application ~~applications~~ in response to concurrently provided query results extracted from the data store via streaming as the results are produced, wherein a query is executed in parallel over the network on multiple nodes and the query results are

streamed provided to the multiple instances of the application applications that are processing in parallel over the network on the multiple nodes and are instances of one another, the multiple instances of the application applications consume the query results to produce the application data, which is streamed as it is produced to then housed in the temporary tables, and wherein when the query is executed the query results are extracted and different portions of the results are streamed provided to each of the multiple instances of the application applications for parallel processing of the results with one another and for producing the application data, each application instance producing a different portion of the application data; and

an application data table that houses the application data once the applications have finished producing the application data, and wherein the temporary tables are merged into the application data table after each of the multiple instances of the application applications have finished processing and populating ~~populate~~ the application data table with the application data, the application data represents processing results of the multiple instances of the application applications produced in response to the query results and the application data is housed in the application data table of the data store for subsequent access.

21. (Currently Amended) The data store of claim 20 wherein a merge utility merges the temporary tables to produce the application data table once each of the ~~plurality of~~ multiple instances of the application applications have finished processing the query results.

22. (Currently Amended) The data store of claim 20 wherein one or more extract utilities perform a query against the data store in order to acquire the query results, which are concurrently consumed by the multiple instances of the application applications to produce the application data.

23. (Previously Presented) The data store of claim 22 wherein each of the one or more extract utilities concurrently populate the query results to application queues.

24. (Previously Presented) The data store of claim 23 wherein each of one or more load utilities concurrently receive portions of the application data from load queues and concurrently populate the portions to the temporary tables.

25. (Original) The data store of claim 22 wherein the data store is a least one of one or more databases and a data warehouse.